UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/752,243	12/28/2000	Nicholas G. Samra	2207/10613	7462

7590 08/13/2004

KENYON & KENYON
333 W. San Carlos, Street, Suite 600
San Jose, CA 95110-2711

EXAMINER	
MEONSKE, TONIA L	
ART UNIT	**PAPER NUMBER**
2183	

DATE MAILED: 08/13/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/752,243	**Applicant(s)** SAMRA, NICHOLAS G.	
	Examiner Tonia L Meonske	**Art Unit** 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 May 2004*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)
2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.
4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
5) □ Notice of Informal Patent Application (PTO-152)
6) □ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1, 2, 9-12, 14, 16, 17, 19, and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith et al., US Patent 5,367,651, in view of Alpha Architecture Handbook (Herein after referred to as "Alpha"), and Colwell et al., US Patent 5,524,262.

3. Referring to claim 1, Smith et al. have taught a zero-generating apparatus for use with an instruction set architecture without an r0 register (Smith does not contain a r0 register.), for detecting a zeroing instruction (Figure 7, 86b-clear register, 86d-clear register, column 5, lines 44-57), deleting the zeroing instruction (column 5, lines 44-57, Figure 7, The clear register instruction is deleted.), and modifying the subsequent instruction to account for the deleted zeroing instruction (Figure 7, column 5, lines 44-57, The load byte instruction is modified to the load byte zero extended instruction.).

4. Smith et al. have not specifically taught a physical zero register which reads as zero. However, Alpha has taught a physical zero register which reads as zero (Alpha, page 3-1, section .1.2 entitled "Integer Registers", R31) for the desirable purpose of eliminating the need for a separate zeroing instruction, which in turn speeds up execution time. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the invention of Smith et al. include a physical zero

register which reads as zero, as taught by Alpha, in order to eliminate the need for an extra zeroing instruction.

5. Smith et al. has also not taught a Register Alias Table (RAT) for storing an instruction register map. However, Colwell et al. have taught a RAT for storing instruction register maps for providing register renaming to provide a large physical register set than would ordinarily be available to eliminate false data dependencies and reduce overall performance of a microprocessor (Colwell et al., abstract, column 3, lines 13-44, column 3, lines 63-column 4, line 7). Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the apparatus, as taught by Smith et al., include a Register Alias Table (RAT) for storing an instruction register map for providing register renaming to provide a large physical register set than would ordinarily be available for the desirable purpose of eliminating false data dependencies that reduce overall performance of a microprocessor (Colwell et al., abstract, column 3, lines 13-44, column 3, lines 63-column 4, line 7).

6. Smith et al. have also not specifically taught modifying said RAT with a pointer to said physical zero register. However, in order to make use of the functionality of the zero register, as taught by Alpha, the RAT of Colwell et al. must update the pointer for the modified instruction of Smith et al. to point to the physical zero register. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the invention, as taught by Smith et al., Alpha, and Colwell et al., modify the RAT with a pointer to said physical zero register, in order to benefit from the benefit from the physical zero register so that the zeroing instructions can be completely eliminated.

7. Referring to claim 2, Smith et al., in combination with Alpha and Colwell et al. have taught an apparatus in accordance with claim 1, as described above, and wherein:

a. said physical zero register is a read only memory (ROM) (Alpha, r31 always reads as the value zero. You can write to R31, but its value can never be changed from zero. Therefore r31 is a read only memory.).

8. Referring to claims 9-12, Smith et al. have not specifically taught wherein said zeroing instruction is an exclusive or, a subtraction, a multiply, or a move instruction. However, it's obvious that the zeroing instruction could be any instruction, such as an exclusive or, a subtraction, a multiply, and a move instruction, which yields the logical equivalent of zero in a destination register. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the zeroing instruction of Smith et al. be an exclusive or, a subtraction, a multiply, or a move instruction, that yields a value of zero in the destination, as they are all logically equivalent instructions.

9. Referring to claim 14, Smith et al. have taught a Zeroing Instruction Logic (ZIL) unit for reading a plurality of instructions and detecting and modifying a zeroing instruction within said plurality of instructions (Figure 7, 86b-clear register, 86d-clear register, column 5, lines 44-57, The clear register instruction to deleted.); where said ZIL unit modifies instructions dependent on said deleted zeroing instruction (Figure 7, column 5, lines 44-57, The load byte instruction is modified to the load byte zero extended instruction.).

10. Smith et al. have not specifically taught a physical zero register which reads as zero. However, Alpha has taught a physical zero register which reads as zero (Alpha,

page 3-1, section :1.2 entitled "Integer Registers", R31) for the desirable purpose of eliminating the need for a separate zeroing instruction, which in turn speeds up execution time. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the invention of Smith et al. include a physical zero register which reads as zero, as taught by Alpha, in order to eliminate the need for an extra zeroing instruction.

11. Smith et al. have not taught where said ZIL unit detects said zeroing instruction and sets a pointer to said physical zero register in place of said deleted zeroing instruction. However, Colwell et al. have taught a RAT for storing instruction register maps for providing register renaming to provide a large physical register set than would ordinarily be available to eliminate false data dependencies and reduce overall performance of a microprocessor (Colwell et al., abstract, column 3, lines 13-44, column 3, lines 63-column 4, line 7). Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the apparatus, as taught by Smith et al., include a Register Alias Table (RAT) for storing an instruction register map for providing register renaming to provide a large physical register set than would ordinarily be available for the desirable purpose of eliminating false data dependencies that reduce overall performance of a microprocessor (Colwell et al., abstract, column 3, lines 13-44, column 3, lines 63-column 4, line 7). Furthermore, in order to make use of the functionality of the zero register, as taught by Alpha, the RAT of Colwell et al. must set a pointer to said physical zero register in place of said deleted zeroing instruction. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the invention, as taught by Smith et al., Alpha, and Colwell

et al., set a pointer to said physical zero register in place of said deleted zeroing instruction, in order to benefit from the benefit from the physical zero register so that the zeroing instructions can be completely eliminated.

12. Referring to claim 16, Smith et al. have taught an apparatus in accordance with claim 14, as described above and wherein said ZIL unit modifies instructions dependent on said deleted zeroing instruction with a renameable pointer (The instruction dependent on the deleted zeroing instruction, as taught by Smith et al., is modified with a renameable pointer, as taught by Colwell et al., in order to eliminate false data dependencies and reduce overall performance of a microprocessor.).

13. Referring to claim 17, Smith et al. have taught a method of zero-generating with an instruction set architecture without an r0 register (Smith does not contain an r0 register.), comprising:

b. detecting a zeroing instruction (Figure 7, 86b-clear register, 86d-clear register, column 5, lines 44-57);

c. deleting the zeroing instruction (column 5, lines 44-57, Figure 7, The clear register instruction is deleted.);

d. identifying a subsequent instruction using said zeroing instruction (Figure 7, 86b-load byte, 86d-load half-word, column 5, lines 44-57); and

e. modifying said subsequent instruction (Figure 7, column 5, lines 44-57, The load byte instruction is modified to the load byte zero extended instruction.).

14. Smith et al. have not specifically taught a physical zero register which reads as a zero value.

15. However, Alpha has taught a physical zero register which reads as zero (Alpha, page 3-1, section .1.2 entitled "Integer Registers", R31) for the desirable purpose of eliminating the need for a separate zeroing instruction, which in turn speeds up execution time. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the invention of Smith et al. include a physical zero register which reads as zero, as taught by Alpha, in order to eliminate the need for an extra zeroing instruction.

16. Smith et al. have not taught modifying said subsequent instruction with a pointer to a physical zero register which reads as a zero value. However, Colwell et al. have taught a RAT for storing instruction register maps for providing register renaming to provide a large physical register set than would ordinarily be available to eliminate false data dependencies and reduce overall performance of a microprocessor (Colwell et al., abstract, column 3, lines 13-44, column 3, lines 63-column 4, line 7). Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the apparatus, as taught by Smith et al., include a Register Alias Table (RAT) for storing an instruction register map for providing register renaming to provide a large physical register set than would ordinarily be available for the desirable purpose of eliminating false data dependencies that reduce overall performance of a microprocessor (Colwell et al., abstract, column 3, lines 13-44, column 3, lines 63-column 4, line 7). Furthermore, in order to make use of the functionality of the zero register, as taught by Alpha, the RAT of Colwell et al. must set a pointer to said physical zero register in place of said deleted zeroing instruction. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the invention, as

taught by Smith et al., Alpha, and Colwell et al., modifying said subsequent instruction with a pointer to a physical zero register which reads as a zero value, in order to benefit from the benefit from the physical zero register so that the zeroing instructions can be completely eliminated.

17. Referring to claim 19, Smith et al., have taught a method of claim 17, as described above, and wherein: modifying said subsequent instruction involves replacing instruction sources (Figure 7, column 5, lines 44-57, Figure 8, column 8, line 49-column 9, line 8, Sources are replaced when the load byte instruction is modified to the load byte zero extended instruction.).

18. Referring to claim 20, Smith et al., have taught a method of claim 17, as described above, and wherein: modifying said subsequent instruction involves using a move (MOV) instruction (Figure 7, 86b-load byte zero extended, 86d-load half-word zero extended, The load instructions are move instructions.).

19. Claims 3-8, 13, 15, 18, and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith et al., US Patent 5,367,651, in view of Alpha Architecture Handbook (Herein after referred to as "Alpha"), Colwell et al., US Patent 5,524,262, and Rotenburg et al., Trace Cache: A Low Latency Approach to High Bandwidth Instruction Fetching.

20. Referring to claim 3, Smith et al. have taught an apparatus in accordance with claim 1, as described above. Smith et al. have not specifically taught wherein said ZIL unit detects said zeroing instruction in a trace cache line. However, having the apparatus, as taught by Smith et al., implemented in a trace cache processor, where the ZIL would inherently detect said zeroing instruction in a trace cache line, allows the apparatus to

employ aggressive techniques to exploit instruction level parallelism (Rotenburg, page 24, section 1 entitled "Introduction", page 25, section 1.1 entitled "The trace cache"). Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the ZIL unit, as taught by Smith et al., detect said zeroing instruction in a trace cache line, as taught by Rotenburg, for the desirable purpose of employing aggressive techniques to exploit instruction level parallelism (Rotenburg, page 24, section 1 entitled "Introduction", page 25, section 1.1 entitled "The trace cache")..

21. Referring to claim 4, Smith et al. have taught an apparatus in accordance with claim 3, as described above and further comprising: an r0 register field logically coupled to said trace cache line for mapping to said physical zero register (For the above combination of references to work as explained above, the R31 of Alpha register must be coupled to the trace cache line of Rotenburg for mapping to said physical zero register R31.).

22. Referring to claim 5, Smith et al. have taught an apparatus in accordance with claim 3, as described above, and wherein said RAT and said trace cache line are logically coupled to a renaming unit for maintaining said pointer to said physical register (For the above combination of references to work as explained above, then the RAT and said trace cache line must be logically coupled to the renaming unit for maintaining the pointer to the physical register.).

23. Referring to claim 6, Smith et al. have taught an apparatus in accordance with claim 3, as descried above. Smith et al. have not specifically taught wherein said ZIL unit deletes said zeroing instruction from said trace cache line. However, Smith et al. have taught deleting the zeroing instruction (column 5, lines 44-57, Figure 7, The clear

register instruction is deleted.); identifying a subsequent instruction using said zeroing instruction (Figure 7, 86b-load byte, 86d-load half-word, column 5, lines 44-57); and modifying said subsequent instruction (Figure 7, column 5, lines 44-57, The load byte instruction is modified to the load byte zero extended instruction.). Since the zeroing instruction is no longer an instruction to be executed, as Smith et al, have deleted the instruction, it follows that the ZIL needs to delete the zeroing instruction in the trace line as the zeroing instruction is no longer executed. Therefore it would have been obvious to one of ordinary sill in the art at the time the invention was made to delete the zeroing instruction in the trace line as the instruction is not executed and should not be included in the trace of instructions to be executed.

24. Referring to claim 7, Smith et al. have taught an apparatus in accordance with claim 6, as described above and wherein said ZIL unit modifies a subsequent instruction (Figure 7, column 5, lines 44-57, The load byte instruction is modified to the load byte zero extended instruction.), where said subsequent instruction is logically coupled to said zeroing instruction within said trace cache line (For the above combination of references to work as explained above, the subsequent instruction of Smith et al. must be logically coupled to said zeroing instruction within said trace cache line.).

25. Referring to claim 8, Smith et al. have taught an apparatus in accordance with claim 7, as described above, and wherein said ZIL unit modifies said subsequent instruction with an immediate source of zero (The subsequent instruction of Smith et al. is modified with the immediate source of zero in R31 of Alpha.).

26. Referring to claim 13, Smith et al. have taught an apparatus in accordance with claim 7, as described above and wherein said ZIL unit transforms said subsequent

instruction to a MOV instruction (Smith et al., Figure 7, column 5, lines 44-57, The load byte instruction is modified to the load byte zero extended instruction, or a MOV instruction.).

27. Referring to claim 15, Smith et al. have taught an apparatus in accordance with claim 14, as described above. Smith et al. have not specifically taught wherein said ZIL unit modifies instructions dependent on said deleted zeroing instructions with an immediate source of a value when both occur with a single trace cache line. However, having the apparatus, as taught by Smith et al., implemented in a trace cache processor would allow the apparatus to employ aggressive techniques to exploit instruction level parallelism (Rotenburg, page 24, section 1 entitled "Introduction", page 25, section 1.1 entitled "The trace cache"). Furthermore, one would want to optimize the two instructions of Smith et al. in all cases, including when the instructions appear in a singe trace cache line. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the ZIL unit of Smith et al. modify instructions dependent on said deleted zeroing instructions with an immediate source of a value when both occur with a single trace cache line, in order to employ aggressive techniques to exploit instruction level parallelism when both instructions appear in a single cache line (Rotenburg, page 24, section 1 entitled "Introduction", page 25, section 1.1 entitled "The trace cache").

28. Referring to claim 18, Smith et al. have taught a method in accordance with claim 17, as described above. Smith et al. have not specifically taught pointing to a physical zero register where said subsequent instruction is not within a common trace cache line. However, having the apparatus, as taught by Smith et al., implemented in a trace cache

processor would allow the apparatus to employ aggressive techniques to exploit instruction level parallelism (Rotenburg, page 24, section 1 entitled "Introduction", page 25, section 1.1 entitled "The trace cache"). Furthermore, one would want to optimize the two instructions of Smith et al. in all cases, including when said subsequent instruction is not within a common trace cache line. Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the invention of Smith et al. point to a physical zero register where said subsequent instruction is not within a common trace cache line in order to employ aggressive techniques to exploit instruction level parallelism when both instructions are not within a common trace cache line (Rotenburg, page 24, section 1 entitled "Introduction", page 25, section 1.1 entitled "The trace cache").

Referring to claim 21, Smith et al. have taught a method in accordance with claim 17, as described above. Smith et al. have not specifically taught wherein said subsequent instruction is modified in response to its location in a trace cache relative to said zeroing instruction. However, having the apparatus, as taught by Smith et al., implemented in a trace cache processor would allow the apparatus to employ aggressive techniques to exploit instruction level parallelism (Rotenburg, page 24, section 1 entitled "Introduction", page 25, section 1.1 entitled "The trace cache"). Furthermore, since the zeroing instruction is no longer an instruction to be executed, as Smith et al. have deleted the instruction, it follows that the ZIL needs to delete the zeroing instruction in the trace line as the zeroing instruction is no longer executed. Therefore it would have been obvious to one of ordinary sill in the art at the time the invention was made to delete the zeroing instruction in the trace line as the instruction is not executed and should not be included in the trace of instructions to be executed. As a result, the subsequent instruction must be modified in response to its location in a trace cache relative to said zeroing instruction in order to delete out the zeroing instruction from the trace cache line and move up the location of the subsequent instruction in the trace cache.

Response to Arguments

29. Applicant's arguments with respect to claims 1-21 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

30. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tonia L Meonske whose telephone number is (703) 305-3993. The examiner can normally be reached on Monday-Friday, 8-4:30.

31. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie P Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

32. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

tlm

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100